A Message-Switched Architecture for Challenged Internets

Kevin Fall

IRB-TR-02-010

July, 2002

Intel Research Berkeley

A Message-Switched Architecture for Challenged Internets

Kevin Fall
Intel Research, Berkeley

ABSTRACT

The highly successful architecture and supporting protocols of today's Internet operate poorly when faced with operating environments such as very long delays, excessive loss, network partitioning, or short node lifetimes. Such properties are typically found in extreme environments which lack infrastructure or operate under severe constraints such as limited power. To achieve interoperability between such networks, a network architecture and application interface structured around message switching, one-way delivery paths, and limited expectations of end-to-end contemporaneous delivery is suggested. The architecture is intended to operate as an overlay above the transport layers of the networks it interconnects, and provides key services such as in-network retransmission, interoperable naming, authenticated forwarding and a coarse-grained class of service delivery.

1.0 Introduction

The existing TCP/IP based internet operates on a principle of providing end-to-end inter-process communication through a concatenation of dissimilar link-layer technologies. End-to-end connectivity is enabled by the standardization of the IP protocol and its mapping into link-layer data frames at each router as required. Although often not explicitly stated, a number of key assumptions are made regarding the overall performance characteristics of the underlying links in order to achieve smooth operation: an end-to-end path exists between a data source and its peer, the maximum round-trip time between any node pairs in the network is not excessive, and the end-to-end path loss probability is small. Unfortunately, a class of so called *challenged networks*, which may violate one or more of the assumptions, are becoming important and may not be well served by the current end-to-end TCP/IP model.

Given the large accumulated experience and number of systems compatible with the TCP/IP protocols, it is natural to apply the highly successful architectural concepts of the Internet architecture to new or unusual environments. Such environments, including space or ocean (acoustic underwater) communications, sensor/actuator networks, and military tactical communications share the lack of infrastructure as a common feature. In this paper, we suggest a message-switched overlay architecture as the appropriate approach to tie together such networks to form an "internetwork of challenged internets". The approach is influenced by both the classical ARPANET design and the US Postal System, each of which have become successful communication networks in their own right.

2.0 Challenged Networks

Qualitatively, challenged networks (or perhaps more correctly "performance-challenged internetworks") include internetworks in which end-to-end latency, bandwidth asymmetry, message or bit loss probability, node longevity, or best path stability are substantially worse than is typical of today's Internet. Quantitatively, we use the following metrics to help define the notion of a challenged network. (The following assumes a message m of size $|m|$ bytes is to be sent over a cascade of i links. Each link is parameterized with transmission rate R_i, propagation delay p_i, processing delay Z_i and queuing delay q_i). The processing delay may include time to perform coding or other data-dependent operations, and the queuing delay is a function of time:

1. end-to-end (one-way) latency = $\sum_i \frac{|m|}{R_i} + p_i + q_i(t) + Z_i(m)$ (abbreviated OTT in table below)
2. raw bandwidth asymmetry = $min(R_i)$ forw / $min(R_i)$ back
3. $Pr_{success} = [(1-p_e)^{i(8m)}]$ for cascaded links with IID constant BER p_e

	Wired Internet	Internet (Satellite)	Oceanic Acoustic	Deep Space
e2e OTT	< 2s	< 5s	< 1min	> 4 min
assym	< 30	< 300	1	to 2000
loss	< 5%	< 30%	< 10%	< .01%

TABLE 1. Typical path performance in "challenged" networking environments

Table 1 summarizes typical bounds for these metrics seen by users in a number of unusual network settings. The data for this table represents a collection from a number of sources [Pax97, DMT96, ARINC, KB00]. The strikingly low loss rate for deep space links is due to extensive use of FEC and high power levels.

2.1 High Latency

End-to-end latency represents the sum of one-way delivery delays at each hop and comprises transmission/processing/propagation time across each link, plus any queuing delay experienced during forwarding. Generally, high latency limits the performance of applications and protocols which include some form of adaptation to network conditions. Protocols most severely affected are those which utilize closed control loops to adjust their data sending rates or require timely dissemination of state to accomplish a distributed computation. For example, constructs such as positive acknowledgements (ACKs) or negative acknowledgements (NACKs) with flow control information are used extensively for controlling data retransmission and limiting network or end-node congestion in reliable transport protocols. When faced with high latencies, the data sender is not able to quickly determine what data has been correctly received at the receiver or adjust its sending rate to efficiently use available network bandwidth without congestion. High latencies may affect the performance of distributed algorithms by inhibiting the global sharing of consistent system state. As a result, a distributed computation may fail to converge to a common solution, resulting in inconsistent results or oscillatory behavior. As a specific example, without timely flooding of topology information, typical link-state routing protocol computations will be unable to compute a consistent routing graph when network topology changes at a rate exceeding the topology flooding time.

The transmission delay for a network packet on single link is generally taken to be the packet size (in bits) divided by the channel rate (in bits/sec), a computation which effectively considers the processing delay to be negligible. This transmission time is sometimes compared against the link propagation time to produce the "A ratio" (A), defined as the ratio of the propagation time to transmission time. For most terrestrial wired and wireless links, $A < 1$. However, for several specialized links (and some high-speed conventional links), $A >> 1$. In the case of challenged networks (or links), the signal propagation time may be comparatively long due to either the physical separation between the communicating endpoints (e.g. space) or because of relatively long signal propagation time in the transmission media of interest (e.g. acoustic communication in air or water).

For multi-hop paths in packet networks with statistical multiplexing, the queuing time generally dominates the transmission and propagation delays. For conventional packet networks, queuing time rarely exceeds a second or two, but in challenged networks where no end-to-end path is currently available (see below), the queuing time could be extremely large (hours, perhaps days).

2.2 Bandwidth Asymmetry

Bandwidth asymmetry arises due to routing path asymmetry, different forward/reverse path link technologies, or intentional engineering trade-offs. The *raw bandwidth asymmetry* metric measures the ratio of forward-path minimum capacity versus reverse-path minimum capacity. Not all challenged networks have a large bandwidth asymmetry, but the important class of specialized remote devices often do. For example, remote instruments placed in extreme environments (space, deep water, etc) are often used for some form of telepresence where a comparatively low-bandwidth control channel is paired with a high-bandwidth telemetry channel to retrieve sensor and/or multimedia data such as moving images. For such devices, the bandwidth asymmetry ration can be in excess of 1000. The trend is toward increasing bandwidth asymmetry as inexpensive higher-resolution cameras and sensors become available.

Bandwidth asymmetry adversely affects some reliable protocols by altering the ACK or NACK return path which is used to control the timing of retransmissions. This effect has been studied extensively with TCP, and is discussed below in Section 3.1. At a higher layer, request/response applications that have not been appropriately tuned to balance the amount of request versus response traffic can time out waiting for data to travel across the congested lower-capacity link. Although system interface extensions may help some applications estimate an appropriate choice for retransmission interval, such extensions are not widely available or standardized, and in the most extreme case of bandwidth asymmetry, are not useful (when communicating one-way to submarines, for example). In this case, use of error correcting codes and non-adaptive periodic retransmission are typically used. See [BLMR98] as an example of how such techniques are used in the Internet.

Moderate asymmetries in the wired Internet are found most commonly in asymmetric access technologies such as in CATV or DSL-based subscriber lines. The largest asymmetries for wireless Internet access appear to be emerging in in-flight Internet access systems. The AirTV system, for example, being proposed for 2004 and beyond, could provide a ground-to-air speed of up to 40Mb/s from each of four satellites using DVB satellite technology with a comparatively anemic air-to-ground bandwidth of a 400kb/s or less using INMARSAT [ARINC].

2.3 Packet Loss and Bit Errors

The probability of success metric given above is the probability of successful end-to-end delivery of a particular message of size $|m|$ bytes across i links assuming an IID stationary geometric loss process with constant bit error rate p_e across all intervening forwarders. While the simple model of loss may not be entirely realistic, the fundamental problem of loss remains even with other models. The loss metric, as expressed, does not account for congestive loss but does account for message size and number of cascaded links given a fixed BER. For packet networks, most links employ some form of error detection, implying that any bit error creates an end-to-end packet loss. As can be clearly seen from the formula, the end-to-end probability of successful delivery decays exponentially with path hop count. Any congestive loss would only worsen the performance, but some of these networks are engineered with admission control or channel reservation so as to effectively eliminate loss due to in-network congestion.

For reliable transfer, excessive errors require repair using either error correcting codes or retransmission. In the case of end-to-end retransmission, the path can be so lossy as to effectively cause end-to-end retransmission to be useless. Given the assumptions listed above, the expected number of retransmissions required before successful delivery is given by: $(1-(1-p_e)^i)/(1-p_e)^i$. Considering an error probability of 0.3. In this case, 4 hops requires 3 retransmissions, 10 hops requires 34, and 20 hops requires about 1200 retransmissions. If a hop-by-hop retransmission scheme were used instead, the total number (network wide) number of retransmissions is given by $ip_e/(1-p_e)$. For the same error probability of 0.3, the number of retransmissions for 4, 10, and 20 hops would be 2, 5, and 9, respectively. Thus, for very lossy environments, an end to end retransmission strategy will not provide satisfactory performance.

2.4 Defining a Challenged Internet

The types of networks falling in the challenged category presented here are generally those which deviate significantly from the performance experienced in the Internet. “Significantly” can be defined informally as anything an order of magnitude larger (or smaller) than the comparable metric in the Internet. This places a number of the networks in Table 1 into the challenged area, but for different reasons. For some wireless networks (e.g. tactical military), high loss rates, long queuing delay (due to competition from telecom traffic) and mobility can lead to significantly different performance than experienced in the wired (or LAN-based wireless) Internet. For acoustic-based networks in water, the channel provides up to about 15KHz of bandwidth with a speed-of-sound propagation of about .67s/km and typical communication distances of up to 5km using acoustic modems. This slow signal propagation rate, in combination with errors created from interference of various natural and man-made sources, makes this type of network challenged from both a latency and error point of view.

Frequently, challenged networks comprise a small number of specialized communication links designed without internetworking in mind. Indeed, most of the research energy devoted to these types of networks goes into link engineering. Without direct attention to internetworking *per se*, such networks are typically interfaced with the Internet on a per-application basis using proxies, if at all. Although this approach is workable, it leads to an unsystematic col-

lection of point networks. Without a common protocol to interconnect them, these networks do not adopt a common set of interfaces or services, implying their users must contend with peculiarities of naming and addressing, message format, security and other issues on an *ad hoc*, per network basis. In addition, without a routing function supported between proxies, selecting which proxies to use for inter-network communication is especially difficult. For these reasons, we believe a network architecture and supporting protocols spanning these types of networks is warranted. The first question to answer is whether the existing Internet protocols and application interfaces could be used as the solution.

3.0 Problems with the Internet Protocols and Applications

3.1 The Core Protocols

The performance characteristics of challenged networks contribute to confound the efficient operation of the core Internet protocols. By the 'core' Internet protocols we mean IP, TCP, UDP, BGP, common IGPs (RIP, OSPF, or EIGRP) and DNS. These protocols span the services of end-to-end datagram delivery, reliable two-party stream delivery, regional (aggregated) routing path discovery with policy, intra-domain path selection and distributed support for name resolution. Although some of these protocols are technically "application" protocols from a layering point of view, we treat them here together as core protocols for the purposes of discussion.

Excessive latency affects TCP directly by severely limiting its throughput performance or interfering with connection establishment [DMT96]. Any application-layer protocol using TCP as its underlying transport is therefore affected (BGP and DNS zone transfers). Excessive latency also adversely affects the proper operation of conventional routing protocols as links will be incorrectly discovered to be non-operating when soft-state refreshes are delayed too long (RIP), or a lack of response to link state "hello" messages (OSPF, EIGRP) is observed. UDP is not sensitive to excessive latency because it does not contain timers that affect its operation, but core application protocols that require it (DNS queries/responses plus SNMP if used) will take too long to complete when faced with excessive delay, triggering early application failure (or the lack of ability to use DNS name/address mappings in the case of address-to-name queries). While application-level timeouts could conceivably be hand-tuned for communicating with particular challenged networks, this approach is unlikely to scale well in the face of large numbers of challenged networks with wide ranging performance capabilities using the Internet for transit.

Bandwidth asymmetry adversely affects TCP by altering the smooth flow of acknowledgments. Extensive studies have been conducted on TCP using the *normalized asymmetry ratio*, which takes into account the asymmetric message sizes between data packets and returning ACKs [MLS00]. Results indicate that bandwidth asymmetry can lead to poor performance of unidirectional transfers due to alterations in the time series of the ACK channel. In particular, if ACKs are queued, their spacing in time causes the TCP sender to clock out subsequent packets less frequently. If ACKs are lost, burstiness, slow congestion window growth, or defeating of the fast retransmit/recovery algorithms can occur. (See [PILC-ASYM] characterization of this problem and some possible approaches to ameliorate it). Any application layer protocols attempting to use TCP over asymmetric links are therefore affected as well.

Significant path loss affects the TCP transport most strongly, causing it a number of problems. After multiple loss events it will continue retrying with a backed-off retransmission timer until it gives up on retransmitting altogether and closes the connection. Somewhat more moderate losses will contribute to problems invoking the fast retransmit and recovery algorithms, even in the presence of selective acknowledgements. IP performance can be affected by path loss if fragmentation is required. IP provides no mechanism for fragment retransmission, thereby causing the overall probability of successful datagram delivery to be further reduced if datagrams are fragmented [M95].

Node longevity affects the probability of end-to-end delivery, as round-trip or even one-way delivery time of a particular message may exceed the sender's lifetime. Clearly, in such cases it is useless to arrange for immediate acknowledgements to be returned. In such cases, any notification of successful or unsuccessful delivery needs to be directed to some alternative node that remains functional.

3.2 Common Internet Applications and their Protocols

The most common user services within the Internet today are the web, e-mail, and perhaps FTP. These services employ a number of different application protocols, including HTTP, SMTP, POP, IMAP, Microsoft's Exchange Protocol and FTP. These protocols, or in some cases the applications that use or implement them, are all of the request/response type involving a client-server type interaction. Generally speaking, they all perform poorly or not at all under the conditions described in Section 2.0. The four most common problems are application-level timeouts mismatched to the actual experienced latency, the inability to automatically choose alternative servers during a failure, a programming interface that assumes application process execution is relatively long compared to transaction duration, and an excessive number of round-trip request/response protocol message exchanges required in order to accomplish a protocol transaction. The first three issues can lead to complete failure, while the last one generally leads to poor performance.

Application-level timeouts are typically used to initiate a transaction retry, or to inform a user that some requested transaction cannot be completed (in the time provided). For example, web browsers and mail clients will often time out during connection establishment after a minute or two, awaiting a response from their peers. These applications make the assumption that a connection should be able to be created in a fairly short amount of time, and that user intervention is a fallback option on failure. These types of applications typically contain no routing function, so any attempt to use an alternative server requires manual reconfiguration.

The assumption that connection establishment and duration will be relatively short-lived compared to the execution time of an application is implicit in the design of the most common network programming interface -- sockets. In particular the port binding functions provided by the `bind()` call are only persistent for at most the duration of the application. In addition, most of the supporting functions (connect, accept, etc) are blocking calls by default, and similarly persist at most as long as the application is in execution.

Electronic mail comes close to addressing the problems posed by the need for delay tolerance. Email falls short most fundamentally due to its lack of dynamic routing. Email generally, and the SMTP protocol particularly, makes use of a statically-defined set of mail exchanger pointers that provide a limited method for using alternative mail drops to deliver mail. Furthermore, the mail architecture as currently conceived does not accommodate a set of intermediate mail redistribution points (message routers). Thus, if an especially lossy path (or no path) exists between a sender's SMTP agent and a receivers mail spool, TCP will be unable to make any significant progress in delivering e-mail toward its destination.

Another particular problem with the SMTP protocol is its startup sequence. At the beginning of an e-mail exchange, each peer must mutually identify itself prior to actually exchanging the mail payload, even though accomplishing this name exchange early is not fundamental to the process of delivering e-mail. Such protocols have been called "chatty" due to their excessive number of round-trip times required to accomplish a simple task. FTP and TELNET each have a similarly chatty interaction during their initial associations when user login, authentication information, and terminal capabilities are exchanged.

3.3 Discussion

In consideration of using the existing Internet protocols as a basis for interconnecting challenged networks, certain properties of the operational environment seem insurmountable. The possibility that only a one-directional path is available at any given time appears to preclude the use of conventional TCP-like protocols based on ARQ for reliability because an ACK channel may not be available. With exponentially-decaying probability of successful end-to-end delivery as path length grows, end-to-end reliability should be avoided in favor of hop-by-hop retransmission in lossy environments. Given the possibility of very high round-trip times, any expectation of timely response in request/response protocols must also be avoided. Finally, the regular Internet routing protocols assume the immediate existence of an end-to-end path and do not accommodate the notion of future (scheduled) routing opportunities that are needed for operation in frequently-disconnected environments.

4.0 A Message-Switched Overlay Architecture

The architecture proposed for interoperability between and among challenged networks is called the *delay tolerant networking* architecture (DTN), and is based on message switching. Messages are known as "bundles" and are adopted from [IPN]. The use of message switching for non-interactive traffic can provide benefits over packet switching from the network point of view because it allows the network to have *a-priori* knowledge about the size and performance requirements of requested data transfers. An application interface structured around message switching also tends to limit an application's expectation of short request/response interaction times, which is beneficial to enhance the overall system's robustness to high delay. Although virtual circuits also share some of these properties, the need to pre-establish network state end-to-end in such systems is considered an inappropriate match for the link characteristics assumed here.

By combining a-priori knowledge of messages awaiting delivery with network topology and performance information, admission control, storage allocation and message routing and scheduling can be dynamically computed relatively early in the lifetime of a message. By matching pending messages to network capacity (which may be intermittent; see below), more sophisticated path selection algorithms beyond shortest path may allow for the use of multiple delivery paths simultaneously. Also, given an appropriate encapsulation, message switching easily supports multi-message multiplexing or "bundling" together to form an aggregate useful for message scheduling and routing.

DTN is an "overlay" architecture in that it is expected to operate above the existing protocol stacks present in other network architectures. For example, in the Internet the overlay may operate over TCP/IP, in the space context it may operate over CFDP/CCSDS [CCSDS], and in sensor/actuator networks it may operate over a network composed of a yet-to-be-standardized sensor transport protocol with some specialized routing (e.g. Epidemic, Diffusion, DSDV [VB00, IGE00, PB94]). Each of these networking environments have their own specialized protocol stacks and naming semantics developed for their particular application domain. Achieving interoperability between them is accomplished by special DTN gateways located at their interconnection points.

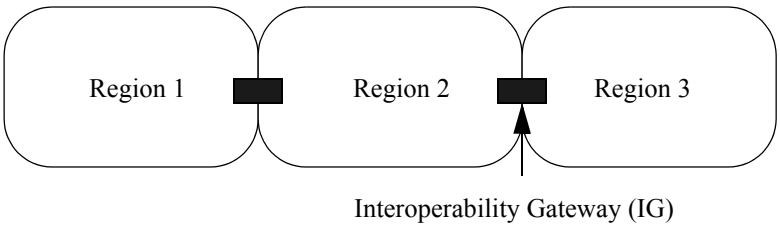

FIGURE 1. Interoperability gateways are DTN forwarders interconnecting regions running potentially incompatible protocol stacks. By operating above the transport protocols in use on the incident networks, they provide message switching, in-network retransmission, and name mapping, allowing the use of globally-interoperable names to be mapped to region-local names as required by the adjacent region's delivery semantics.

4.1 Regions and Interoperability Gateways

The architecture includes the concepts of *regions* and *interoperability gateways*. Region boundaries are defined as interconnection points between dissimilar network protocol and addressing families. We expect a small number of *region types* may evolve and each instance of the same type will implement a similar stack of protocols. Between regions are interoperability gateways (IG), which correspond to both the Metanet "waypoint" concept in [META] and also to the definition of gateways described in the original ARPANET design [CK74]. Nodes within a region can only exchange data with nodes in other regions with help of an IG. The waypoint concept implies a point through which data must pass in order to gain entry to a region. This point can serve as a basis for both translation (between region-specific encodings) as well as a point to assert policy and control.

An IG spanning two regions consists logically of two "halves," each half in one of the adjacent regions, in a fashion similar to an ARPANET-style gateways structured above specific link layer protocols. In operating above the transport layer, however, IGs differ from ARPANET gateways and are not focused on packet switching. They are responsible for message storage and switching between differing transports and mapping globally-significant name tuples to locally-resolvable names for traffic destined internally to an adjacent region (see following section). They also perform the functions of DTN forwarders as defined in section 4.4.

4.2 Naming and Addressing

The factors influencing naming and addressing include: what objects are named (typically nodes or data objects), whether a name can be directly used by a data router in order to determine the delivery path, and the method by which name/object bindings are managed. In the classical operation of the Internet, globally unique destination addresses are compared at each hop versus a forwarding table in order to determine the next-hop address. In the implementation of most network switch nodes today, the next-hop address is internally represented by the underlying address (e.g. MAC address of the next hop). Names are provided by DNS as an application-based overlay, and the network can continue operations, albeit in a degraded mode, even when DNS has failed.

For routing of DTN bundles, we elect to use identifiers for objects or groups of objects called *tuples* comprising two variable length portions. The first portion identifies a region and is interpreted by DTN forwarders to find the path to one or more interoperability gateways at the edge of the specified region, and the second portion identifies a name within the specified region. As a message transits across a potentially long and heterogeneous collection of regions, only its region identifier is used for routing. Once a bundle reaches the edge of the destination region, the name information is locally-interpreted, and translated if necessary, into a name appropriate to the containing region. This method of resolving names results in a form of *late binding* for tuples in which only the portion of the tuple immediately needed for message forwarding (the region portion) is used by DTN nodes. By not imposing any particular fixed structure on the name portion of a tuple, any reasonable naming scheme can be easily accommodated.

DTN name tuples have a structure of the form {R, L} where R is a variable-length, hierarchical region identifier and L contains a name local to region R and treated as opaque data outside region R. In the case of the Internet, for example, we could have the following tuple:

{internet.earth.sol.int, "http://www.ietf.org/overview.html"}

Late binding of tuples in DTN differs from the DNS-style Internet naming and addressing which requires one or more DNS transactions to complete prior to the start of an Internet end-to-end conversation. For challenged networks, the need to consult a name-to-address mapping that may be resident only in the destination region seems impractical given potentially large end-to-end delays. While it could be argued the DNS naming *structure* can possibly be separated from its implementation (thereby eliminating the request/response round-trip required to execute the DNS distributed database access), the DNS structure does not have any explicit method for handling completely opaque naming data or late binding.

The choice of adopting names rather than addresses as the basis for identifying objects derives from an observation of recent trends in the operation of the Internet. The Internet design makes frequent reference to resource sharing as enabled by a (distributed) interprocess control mechanism. Addressing is used for routing and referring to a computational resource, and naming is applied to make the addressing easier for humans. Today's Internet includes objects such as search engines and page caches which are used extensively. In many cases, a name (generalized to a URL or URI) effectively refers to a query for data rather than identification of a particular end-system computational resource. Not surprisingly, this fact is being worked into the overall network architecture in a number of recent studies in "data-centric" naming schemes. See [SMKKB01] as one such example.

Option Name	Mailing Receipt	Delivery Record	Air Delivery	Recipient Pays	Moves Money	Delivery Confirm	Return Receipt	Careful Handling	Insurance	Restricted Delivery	Signature Confirm
Cert. Of Mailing (RM)	Y		(w/PAL)					(w/SH)			
ParcelAirLift (PAL)			Y								
Special Handling(SH)			(w/PAL)	(w/COD)		(w/DC)	(w/RR)	Y	(w/IM)		(w/SC)
Certified Mail(CM)	Y	Y					(w/RR)			(w/RD)	
COD	(w/RM)	Y		Y		(w/DC)	(w/RR)	(w/SH)	(w/RM)	(w/RD)	(w/SC)
Delivery Confirm(DC)				(w/COD)		Y	(w/RM)	(w/SH)	(w/IM or w/RM)		
Insured Mail(IM)			(w/PAL)			(w/DC)		(w/SH)	Y		(w/SC)
Money Order					Y						
Return Receipt(RR)	Y	Y	(w/PAL)			(w/DC)	Y	(w/SH)		(w/RD)	(w/SC)
Registered Mail(RM)	Y	Y		(w/COD)		(w/DC)	(w/RR)		Y	(w/RD)	(w/SC)
Restricted Delivery(RD)			(w/PAL)			(w/DC)	(w/RR)	(w/SH)		Y	(w/SC)
Sig. Confirm		Y				Y					Y

() - indicates the names of options required for specific services
Y - indicates the option provides the service directly

TABLE 2. US Postal Service Classes of Service

Although the DNS/addressing approach results in greater forwarding efficiency at IP routers (no per-hop address resolution is required), we do not consider the need to perform this sort of lookup detrimental to our approach, as the entire design is not focused on high speed. Indeed, the tuple structure could imply more than two indirect lookups to ultimately determine an endpoint: one in order to resolve the region identifier to a valid next-hop, and a second lookup to resolve the region-specific data to a valid next-hop or aggregate set within the specified region. Importantly, however, such queries are limited to the immediate region and do not require a complete end-to-end transaction. In the case of the data-centric naming proposed for the Internet, per-hop name resolution performance would be more important, as these systems aim to present an alternative to current routing techniques which must operate at high speeds.

The existing NS record structure within DNS appears to be sufficient for supporting a mapping between region identifier and a gateway (or "waypoint") from within the Internet. For challenged networks the conventional DNS protocol is not an attractive option due to its request/response operation. In such cases, DNS-like services that do not require request/response interaction over high round-trip-time paths will need to be placed appropriately to allow poorly-connected devices to obtain a useful next-hop destination for its data. Something analogous to a default next hop with late-binding of the routing tag may suffice, but requires further investigation.

4.3 A Postal Class of Service

The notion of a challenged network inherently implies a limitation on various resources. Priority-based resource allocation is therefore important to adopt in the overall model, but care must be taken to avoid so burdensome a class of service architecture as to have it be unimplementable or confusing to users in many cases. The approach taken here is to adopt a subset of the types of services provided by the US Postal Service. This system has evolved to meet the needs of millions of users exchanging non-interactive traffic and has the added benefit of already being reasonably familiar to most users. As such, it seems a highly compelling starting point for considering the classes of service to be offered by a primarily non-interactive networking architecture.

Over hundreds of years, the US Postal System has developed a remarkable class of service offering associated with the apparently straightforward service of mail delivery. In addition to the basic delivery categories of first-class, priority, express mail, parcel post and "bound printed matter", Table 2 above indicates the various special delivery options and the nature of the services they are designed to provide. In this table, the first column indicates the name of the option and possibly its abbreviation, and the first row indicates the intended service. The entries in the matrix indicate if the service is directly supported by the option (indicated as "Y"), or whether it is available in combination

with some other option (indicated parenthetically). Empty entries in the matrix indicate a lack of support for the service using the corresponding option.

As can be seen from the table, some combinations of options are not supported, whereas other options have mutual interdependence. The complexity of this system seems too high as a basis for a network class of service offering, as several of the options are not directly applicable to a data network (e.g. air delivery) or are tied to financial considerations that are considered to be out of scope for the DTN design (e.g. insurance or "moves money"). In a distilled form, however, the following core services seem to be attractive due to their coarse granularity and intuitive character: low, ordinary, and high priority delivery; notifications of mailing, delivery to the receiver (return receipt), and route taken (delivery record). The model is extended with the option of reliable delivery (somewhat akin to careful handling), and messages requiring this service are handled somewhat differently by the routing system in that they require a custody transfer at each routing hop (see Section 4.4.1 below).

4.4 Functions Performed by a DTN Forwarder

A DTN forwarder is primarily responsible for the forwarding of DTN bundles between DTN hops using the underlying transport protocols appropriate to its containing region. Forwarding may involve storage and scheduling of bundles, aggregation of bundles, generation of status messages relating to bundle delivery, enforcement of security policy and participation in network time synchronization. DTN forwarders play the role of regional and local post offices in the postal network.

4.4.1 Path Selection and Scheduling

The DTN architecture is targeted at networks where an end-to-end routing path cannot be assumed to exist. Rather, routes are comprised of a cascade of time-dependent *contacts* (communication opportunities) used to move messages from their origins toward their destinations. Contacts are parameterized by their start and end times, capacity, endpoints, and direction. In addition, a measure of a contact's predictability can help to choose next-hop forwarders for message routing as well as select the next message to be sent. The predictability of a route exists on a continuum ranging from completely predictable (e.g. wired connection or a periodic connection whose phase and frequency are well-known) to completely unpredicted (an "opportunistic" contact in which a mobile message router has come into communication range with another DTN node). Note that the measure of a contact's predictability is sensitive to its direction. For example, a dial-in connection may be completely predictable from the initiator's point of view while being completely unpredicted from the callee's point of view.

The particular details of path selection and message scheduling are expected to be heavily influenced by region-specific routing protocols and algorithms. At this relatively early stage of development, several challenging problems have been identified: determination of the existence and predictability of contacts, obtaining knowledge of the state of pending messages given assumptions of high delay, and the problem of efficiently assigning messages to contacts and determining their transmission order. While very simple heuristics for these problems can be implemented without excessive problems, each issue represents a significant challenge and remains as future work.

4.4.2 Custody Transfer and Transport Protocols

The routing portion of the DTN architecture includes two distinct types of message routing nodes: persistent (P) and non-persistent (NP). P nodes are assumed to contain nontrivial amounts of persistent message store, and NP nodes are not. Unless they are unable or unwilling to store a particular message, P nodes generally participate in *custody transfer* using the appropriate transport protocol(s) of the containing region. A custody transfer refers to the acknowledged delivery of a message from one DTN hop to the next and the corresponding passing of reliable delivery responsibility. The custody transfer concept is fundamental to the architecture in order to combat potentially high loss rates and the corresponding issues associated with end-to-end retransmission described above.

The facilities provided by the transport protocols in use within the regions containing a DTN P node may vary significantly. For example, any transport protocol may or may not offer the following: reliable delivery, connections (with indications of connection failure), flow control, congestion control, and message boundaries. As the bundle forward-

ing function assumes an underlying reliable delivery capability with message boundaries when performing custody transfer, transport protocols lacking these feature must be appropriately augmented. A implementation structure for a bundle forwarder includes a number of transport-protocol-specific *convergence layers* used to add reliability and message boundaries above those transport protocols requiring augmentation. (Note that TCP in the Internet requires augmentation due to its lack of message boundaries). The design and implementation of convergence layers is specific to the transport protocols being augmented, and are therefore beyond the scope of the overall bundle forwarding design described here.

In cases where reliable delivery is provided by an underlying transport, a bundle forwarder need only manage connection state and initiate restarts if a connection is lost. In the case of connection-oriented protocols, detection of a lost connection is generally provided through the application interface (via signals or other errors using the socket interface, for example). In cases where no direct support is provided for detecting failures, the bundle forwarding function may set a coarse-grained timer to re-start message transfers should it be concluded they have failed. This is designed as a fallback measure in cases of underlying communication failure, and is not expected to be an especially efficient mechanism for initiating retransmission.

Setting the coarse-grain retransmission timer will vary depending on the details of the containing region, and thus represents a certain form of layer violation in which the overlay "network" layer is able to be sensitive to underlying "physical" layer properties. In challenged networks, knowledge of some path properties at the forwarding layer appears to be very useful in selecting error control policy. In space communications, for example, communication opportunities and approximate path delays may be known ahead of time due to planetary dynamics. In other cases, alternative means (e.g. based on geographic location coupled with knowledge of the intermediate communications media) may give reasonable loose bounds on what values to use for the timer. Note that given the often-disconnected state of the network, we cannot generally rely on estimates of the current round-trip time as a basis for initiating retransmission.

4.4.3 Time Synchronization

The DTN architecture requires a level of time synchronization between communicating parties on the order of 1ms. This requirement is more burdensome than the requirements IP places on underlying networks in today's TCP/IP Internet (which is essentially none), but timing is such a fundamental service to many distributed applications and is required by the DTN's approach to scheduling, reliable delivery, and security. While this requirement may seem to be an added burden, we believe the problem of time synchronization is not so difficult to solve as to make it optional. Protocols such as NTP [NTP] have provided 1ms accurate time synchronization (or better) within the Internet, and most existing networks for extreme environments provide some means for time synchronization [DSN].

The need for time synchronization is based on several features common to many challenged environments. First, challenged networks are often used to communicate with devices deployed in hostile remote environments. In such cases, remote instruments are often required to collect data and/or position as a function of time and need to be controlled. While the DTN architecture does not strive to support real-time control loops, it does aim to deliver pre-programmed control instructions to be executed at reasonably precise future points in time. Without reasonably accurate time synchronization, post-facto data analysis and pre-programmed control is made considerably more difficult and unreliable. For security (see below), time synchronization is used to guard against replay attack, and thus must be accurate on the order of the one-way latency. For bundle routing, synchronized time is used to remove pending messages from the delivery system when they expire. This feature does not require especially accurate synchronized time, but deviations of more than a few minutes could prove to be problematic.

4.4.4 Security

The security model for the DTN architecture differs somewhat from traditional network security models in that the set of security principles includes the network routers themselves. Most security approaches involve the mutual authentication and private exchange of data between two network users, leaving the intervening network as a non-participant. In the DTN case, we are more interested in verifiable access to the carriage of traffic at a particular class of service and want to avoid carrying traffic potentially long distances that is later found to be prohibited. To imple-

ment this technique, each message includes an immutable “postage stamp” containing the verified identity of the sender (or role), an approval (and approving authority) of the requested CoS associated with the message, and other conventional cryptographic material to achieve authentication of the message. Routers check credentials at each hop, and discard traffic as early as possible if authentication fails. This approach also has the associated benefit of making denial-of-service attacks considerably harder to mount as compared with conventional Internet routers.

The current approach uses public key cryptography as a starting point for keying. Routers and end-users are issued public/private keypairs, and a user must obtain a signed copy of its public key from a DTN certificate authority. (All routers are assumed to be pre-equipped with copies of one or more DTN certificate authority public keys). The user then presents the signed key along with a message to be carried. At the first DTN router, the signed public key is used to validate the sender and requested CoS. Valid messages are then re-signed in the key of the router for transit. Using this approach, only first-hop routers need cache per-user certificates, and then only for adjacent users. Non-edge “core” routers can rely on the authentication of upstream routers to verify the authenticity of messages. We believe this approach will help to improve the scalability of key management for these networks, as it will limit the number of cached public key certificates to a function of the number of adjacent routers rather than the number of end-users. This should provide both the obvious advantage of space savings, but also an improvement to system management as router keys are expected to be changed less frequently than end-user keys. As DTN routers are likely to be deployed in remote areas, re-keying operations may be a comparatively burdensome system management tasks, so limiting the number and frequency of certificate updates should provide additional savings.

5.0 Application Interface

As described, the DTN architecture is built as an overlay network using messages as the primary unit of data interchange. Applications making use of the architecture must be careful not to expect timely responses and must generally be capable of operating in a regime where a request/response turn-around time exceeds the expected longevity of the client and server processes. In addition, applications must be prepared to handle the creation and manipulation of name tuples and their registrations (for demultiplexing received messages), class of service specifiers, and authentication information. Bundle forwarders must implement the application interface, along with bundle routing and forwarding functionality, persistent message storage facilities, and transport protocol convergence layers. The application interface is non-blocking, and involves registration and callback functions between bundle-based applications and the local bundle forwarding agent. Registration data is persistent, and bundle meta-data is handled with database semantics. Generally speaking, the system should operate without trouble in the face of reboots or network partitioning.

A prototype bundling implementation (running on Pentium and Strong-ARM based Linux systems) has been created, which implements the application interface, rudimentary bundle forwarding, detection of new and lost contacts, and two convergence layers (involving TCP/IP as well as an interface to the Berkeley mote network [MOTE]). Important next steps in the implementation are to include more sophisticated bundle routing function and to implement the security model described above.

6.0 Conclusion

The DTN architecture aims to address the desire to provide interoperable communications between and among a wide range of networks which may have exceptionally poor and disparate performance characteristics. The design embraces the notions of message switching with in-network retransmission, late-binding of names, and routing tolerant of network partitioning to construct a system better suited to operations in challenged environments than most other existing network architectures, particularly today’s TCP/IP based Internet.

The architecture represents a generalization of the Interplanetary Internet architecture described in [IPN] to challenged networks other than space. The previous work was closely tied to issues of deep space communications in particular, but contributed many key ideas toward the development of a networking architecture applicable for challenged internetworks more generally. The design also derives in part from some interesting trends in the Internet: a

move toward content-based naming, creation of administrative “regions”, and alternative routing structures (e.g. network overlays).

The proposed DTN architecture advocates a change to the basic service model and system interface most Internet-style applications have become accustomed to, motivated by the exceptionally poor performance present in some networks. This is a comparatively radical approach; other approaches aim to “repair” underlying link impairments or alter limited portions of the Internet architecture, such as routing, with additional protocols in an effort to keep the current service model and existing TCP/IP based protocols constant. While some of these approaches are performance optimizations (e.g. link-layer FEC or Berkeley’s SNOOP protocol [SNOOP]) and therefore do not obviously negatively impact upper-layer protocols, other approaches such as Mobile IP [MIP] or MANET [AH] imply significant changes to the present datagram delivery system, and are likely to cause more problems with application interaction. Furthermore, these approaches deal primarily with error control and mobility, and not with network paritions or very large latencies. Only time will tell what application interfaces and service semantics will most appropriately match applications to challenged networks, but we believe it is important to consider the full range of options prior to concluding that any new proposed architecture must adopt the semantics of the today’s existing systems.

7.0 Acknowledgement

The author wishes to thank the members of the Interplanetary Internet Research Group for their previous work on the initial definitions of bundling and naming. Members of this group include Vint Cerf (MCI/WorldCom), Adrian Hooke and Scott Burleigh (NASA/JPL), Bob Durst and Keith Scott (the MITRE Corporation), plus Howard Weiss (SPARTA). The author is especially indebted to Bob Durst for an ongoing collaboration regarding the DTN design. Earlier versions of the paper benefited from the comments of David Culler.

8.0 References

[KB00] Daniel Kilfoyle, Arthur Baggeroer, “The State of the Art in Underwater Acoustic Telemetry”, IEEE Journal of Oceanic Engineering, 25(1), January 2000

[DMT96] R. Durst, G. Miller, E. Travis, “TCP Extensions for Space Communications”, Proc. MOBICOM 1996

[ARINC] Presentation by ARINC (see page at http://www.arinc.com)

[META] John Wroclawski, “White Paper - Workshop on Research Directions for the Next Generation Internet”

[IPN] IPNRG, Interplanetary Internet Architecture Draft (see page at http://www.ipnsig.org).

[Pax97] V. Paxson, “End-to-End Internet Packet Dynamics”, IEEE/ACM Transactions on Networking, 7(3), 3/99

[DSN] NASA’s Deep Space Network, See information page at http://deepspace.jpl.nasa.gov/dsn

[BLMR98] J. Byers, M. Luby, M. Mitzenmacher, A. Rege, “A Digital Fountain Approach to Reliable Distribution of Bulk Data”, SIGCOMM 1998

[MLS00] U. Madhow, T. V. Lakshman, B. Suter, “TCP/IP Performance with Random Loss and Bidirectional Congestion”, IEEE/ACM Transactions on Networking, 8(5), Oct 2000

[PILC-ASYM] H. Balakrishnan, V. Padmanabhan, G. Fairhurst, M. Sooriyabandara, “TCP Performance Implications of Network Path Asymmetry”, IETF draft-ietf-pilc-asym-07, (Work in Progress), Nov 2001

[CCSDS] Space standards are developed through the *Consultative Committee for Space Data Systems*. See current information available at http://www.ccsds.org

[VB00] A. Vahdat, D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks,", Duke Technical Report CS-2000-06, July 2000. http://www.cs.duke.edu/~vahdat/ps/epidemic.pdf

[IGE00] C. Intanagonwiwat, R. Govindan, D. Estrin, “Directed Diffusion: A scalable and robust communication paradigm for sensor networks”, MobiCOM 2000, Aug 2000

[PB94] C. Perkins, P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, SIGCOMM 94

[CK74] V. Cerf, R. Kahn, “A Protocol for Packet Network Intercommunication”, IEEE Trans. on Comm., COM-22(5), May 1974

[SMKKB01] I. Stoica, R. Morris, D. Karger, F. Kaashoek, H. Balakrishnan, “Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications”, SIGCOMM 2001.

[RFC1035] P. Mockapetris, “Domain Names - Implementation and Specification”, IETF RFC 1035, Nov 1987

[M95] J. Mogul, “Fragmentation Considered Harmful”, SIGCOMM 1995.

[MOTE] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister. “System architecture directions for network sensors”. ASPLOS 2000.

[NTP] David Mills, “Network Time Protocol (Version 3) Specification, Implementation and Analysis”, RFC1305

[SNOOP] H. Balakrishnan, S. Seshan, R. Katz, “Improving Reliable Transport and Handoff in Cellular Wireless Networks”, ACM Wireless Networks 14(4), December 1995

[MIP] C. Perkins, *Mobile IP: Design Principles and Practices*, Addison-Wesley, Oct 1997

[AH] C. Perkins, *Ad Hoc Networking*, Addison-Wesley, Dec 2000